UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/614,618	07/07/2003	Kedar Sharadchandra Namjoshi		7483

7590 05/28/2008
Ryan, Mason & Lewis, LLP
Suite 205
1300 Post Road
Fairfield, CT 06824

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
05/28/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/614,618
Filing Date: July 07, 2003
Appellant(s): NAMJOSHI, KEDAR SHARADCHANDRA

Kevin Mason
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 2/29/08 appealing from the Office action mailed 10/31/2007.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

However, based on the explanations by Appellant regarding the USC § 112, 2nd paragraph rejection, the 2nd paragraph Rejection is herein withdrawn.

Claims 8-11 and 21-24 are precluded from any grounds of rejection for containing allowable subject matter.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Ben-Ari et al, "The Temporal Logic of Branching Time", Acta Information 20, 6, 1983, pp. 207-220

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-4, 12-17, 25-26 are rejected under 35 U.S.C. 102(b) as being anticipated by Ben-Ari et al, "the Temporal Logic of Branching Time", *Acta Information 20, 6*, 1983, pg. 207-226, (hereinafter BenAri).

As per claim 1, BenAri discloses a method for reducing a program, M, that preserves (Note: induction proof for a proposition or proposed rules to completeness checking reads on preserving the branch temporal property expressed as automata-type of transitions based on induction proof against proposed conditions or abstract notations to represent state/transitions – sec 5, pg. 220; see *branching time approach, correct termination for ... one possible computation for every input ... study the property one possible computation ... all possible choices* - middle pg. 208) at least one branching time property, f, comprising the steps of:

forming a product of said program, M and said branching time property, f, e.g. *program P, automata, branching time... temporal framework* – middle pg. 208) expressed as an automaton A (e.g. *finite model property,* $\mathcal{U}\mathcal{B}$*... all possible combinations* – bottom, pg. 208 Note: Unified UxB model for branching time reads on product to express branching time

property for a input program with all possible combinations, i.e. expressed as automaton – see *nondeterministic, all paths* - pg. 209, top);

obtaining an abstract domain containing a set of abstract values to generalize possible states of said program and abstract relations that relate said program states to said abstract domain (e.g. states, paths, binary relation - sec 2, pg. 209, bottom; Semantics for $\mathscr{UB}$, pg. 210-211; *Structure triple (S, P, R)* tableau - pg. 214-215; Fig. 1 pg. 217);

computing an abstract program with a reduced number of states (Note: formulation of induction proof via formula, axiom, theorem and lemma using rank and marking reads on eliminating of non-valid proposition or unsound transaction, wherein from the initial number of deterministic and non-deterministic states - in the $\mathscr{UB}$ model – see pg. 209-224 only a certain propositions hold true whereas some will not, i.e. some transaction can be determined as behaving correctly whereas some cannot, thus *reduced number of states*) and an altered version of said branching time property f (e.g. *proposition, rules, structure H ... iff* - pg. 212-214 – Note: altered is construed as the result of providing axiom and conditions via propositional calculus whereby branching property of the original input program has been altered via selected BenAri's formulating in forms of structures, propositions or algorithmic mechanism – Tableaux, Marking, predicate/Lemma, Ranking - in order to validate which branch property expressed as transition will be maintained after soundness is established) using said product (e.g. pg. 210, top; *is satisfiable... rules of inference ...complete axiomatization*, pg. 211, top).

As per claims 2-3, BenAri discloses performing an automated program check (e.g. proof – pg. 214-224; *Completeness,* sec 5, pg. 220); wherein said automated program check is a model

checking step (e.g. *model*- pg. 210, bottom; *limit our models*, pg. 211, bottom; *satisfiability ...prove ... finite model* – pg. 213 bottom; *any model ... Hintikka structure* – pg. 214, bottom).

As per claim 4, BenAri discloses wherein said automated program check is performed for an altered branching time property (branching time – pg. 208, bottom).

As per claims 12-13, BenAri discloses the step of obtaining one or more rank functions and employing (*ranking algorithm*, pg 217-218) said one or more rank functions in an abstract transition relation, R'; the step of obtaining one or more choice predicates (e.g. W1, W2 ... alternative node ... our choices – pg. 218, bottom to pg. 219, top) and employing said one or more rank functions in an abstract transition relation, R'.

As per claim 14, BenAri discloses a system for reducing a program, M, that preserves at least one branching time property, f, comprising: a memory; and a processor operatively coupled to said memory, said processor configured to:

form a product of said program M and said branching time property, expressed as an automaton A;

obtain an abstract domain containing a set of abstract values to generalize possible states of said program and abstract relations that relate said program states to said abstract domain;

compute an abstract program with a reduced number of states and an altered version of said branching time property using said product;

all of which limitations having been addressed in claim 1 above.

As per claims 15-17, and 25-26, these claims correspond to the subject matter of claims 2-4, and 12-13, respectively, hence will incorporate the rejections as set forth therein.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 5-7, 18-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ben-Ari et al, "the Temporal Logic of Branching Time".

As per claim 5, BenAri does not explicitly disclose defining a set of states, S', in said abstract program as S'={overscore (S)} X Q, where S is a set of states in said program, M, and Q is a finite set of states of the automaton A; but according to BenAri disclosing a unified $\mathcal{UB}$ branching system whereby a program M is abstracted into a multitude of represented paths and states (refer to claim 1; *finite* model - pg. 208, bottom; pg. 210, bottom) the cross product concept using a automaton is disclosed, and the abstract program M with S states and Q the automaton represented *finite* states is also strongly suggested. It would have been obvious for one skill in the art at the time the invention was made to implement BenAri cross product of program M and the above automaton via the $\mathcal{UB}$ system so that S' would be a cross product from the set of states in said program M and the finite set of states in said automaton used in said $\mathcal{UB}$ system, because this would enable for each such S' set, applying axioms, proposition using lemma, theorem to inductively prove validity (re claim 1) of some formula or tableau concerning a branching temporal behavior for such S' product, based on BenAri's purpose (see pg. 224) for unifying not only deterministic states (finite state) but also non-deterministic via model checking for completeness against each proposition.

As per claim 6, BenAri does not disclose wherein OR states in said set of states, S', are those states where δ(q, *true*) has the form q1 √ q2, and all other states are AND states, where q are individual states and δ is a transition relation between states. But transitions expressed in finite state machines technologies as perceived via BenAri using operators to represent an OR or AND type of operations (V or V-inverse) of δ transaction to yield a states and to equate this operation in terms of it being true or false was a known concept (see pg. 211-212). In light of possibilities of transitions in a program which mostly include either OR or AND type of path (based on the operators V or V-inverse or ∩), as exemplified by BenAri, it would have been obvious for one skill in the art at the time the invention was made to provide symbolism as represented by the branching time temporal abstracted system by BenAri so that OR transitions has q1 v q2 format and AND operations has the other transition formats.

As per claim 7, Alur does not explicitly disclose wherein an abstract state (t, q^) is in a subset of initial states, I', of the abstract program if there exists s ∈ I for which s. ξ(q^)}t, where s is an individual state, I is a subset of initial states, I, of the program, M, and ξ(q^) is one of said abstract relations. The concept of subset among a superset in view of the automaton-based establishing of axiom, theorem, or lemma is strongly conveyed from BenAri propositions using theorems expressed with ∈ (see pg. 214; pg. 215). Based on the cross product of the program M by an branching time logic represented by $\mathcal{U}\mathcal{B}$ system using an automaton, it would have been obvious for one skill in the art at the time the invention was made to implement the unified system by BenAri so that subset *s* belonging to subset *I* initial states is abstracted via ξ(q^) transition to yield a subset state --abstract state (t, q^) -- belonging to a superset I' of said abstract program; because this is how the transitions and states according to BenAri theorem,

proposed tableau -- which includes Є operators --can be validated and checked for completeness (see pg. 212-224)

As per claims 18-20, refer to the rationale of claims 5-7, respectively.

---

(10) Response to Argument

USC § 102(b) Rejection:

(A) Appellant has submitted that, regarding claims 1 and 14, BenAri does not disclose or suggest *reducing* a program (Appeal Brief pg. 6, li. 21-23). The claim language amounts to three steps: (i) forming a product M expressed as automaton A; (ii) obtaining an abstract domain containing possible states and abstract relations; (iii) computing an abstract program with a reduced number of states. There is no further specifics as to how exactly the 'with reduced number of states' have been achieved from the above steps (i), (ii) and (iii).

(B) For background information, the theoretical computer/software field dealing with finite state machine with concern for program behavior in terms of time complexity, deterministic and non-deterministic aspects of a program in form of mathematical notations has been known as automata methodology using formal methods. Accordingly, intersecting a set of states for program source as input space with a automaton's state spaces to yield a abstract representation of all possible states including deterministic and non-deterministic possibilities of (program behavioral) transitions given such input would fall under the ambit of formalizing such interaction or intersection, otherwise represented as cross-product in formal methods; i.e. cross product of program and automata as is the case from the claim. According to Wikipedia, a transition δ is a cross product of a first or initial set of states (Q) and finite set of a program

symbols (Σ) to yield a second or final set of states (e.g. transition function δ: $Q \times \Sigma \rightarrow Q$ according to <http://en.wikipedia.org/Automata_Theory>); thus, formalizing program behavior from submitting a input program into a automaton entails a plurality of very atomic operation such as the above function called a cross product. Therefore, submitting two spaces (a program and a property space expressed as an automata) as in a formalized intersection as set forth above would yield a formal representation called in automata theory, a cross-product of two state spaces, comprised on instances of very atomic cross product as mentioned above.

(C) Interpretation of what appears to be a cross-product of a program and expression of an automaton to yield an abstract representation of *all possible states* as in (i) and (ii) is as following: one of ordinary skill in the art would apply the formalizing of program behaviors using the automata-based methodology as set forth above, and would construe the reduced number of states in terms of the extent at which all possible states are reduced based on the mechanism, a programmatic or mathematical computation, or algorithmic implementation by which a otherwise large set of transitions can be reduced. Since the claimed 'computing' step is devoid of implementation details, the limitation as to 'with a reduced number of states' appears to be a end result being rather devoid of mechanics by which such end result has been attained. Broadly interpreted, the computing step (iii) amounts to having an abstract representation resulting from (i) and (ii) so that these possible states initially obtained from step (ii) are reduced as in (iii). When the claim does not explicitly relate the 'product' formed in step (i) and the possible states contained in the abstract representation of step (ii), and absent any details about 'altered' or 'reducing', one would not be able to see how based on *altering a version* of a branching property f, a *reduced number of states* is accomplished. Broad reasonable

interpretation for one skill in the art has been applied based on known concept at the time of the invention, as following.

(D) At the time the invention was made, in terms of automata theory, cross-product between a program space and an automaton space would be interpreted as generating all possible states including non-deterministic and deterministic transitions in between states; and using formal methods (e.g. propositional calculus as by BenAri) to find a propositional algorithm by which based on a set of formal annotations/abstracting, branch time complexity expressed in terms of possible transitions can be proven valid by inductive proof, as to evaluate correctness of proposition when the to-be-verified transition is proved true compared to its non-deterministic and un-validated prior state.

For example, BenAri teaches propositional calculus to resolve program time complexity in terms of automata model to validate properties of a program (unified as a model) intersected with its branching time space (see Introduction, pg. 208, 2nd and 5th paragraphs). The purport as to reducing an otherwise very large number of states as per the original cross-product would map to choosing a simpler approach via a selective model (or a best abstract representation or mathematical expression thereof) based on which to provide evaluation of correctness (i.e. induction proof of completeness) regarding all these possible states -- as set forth in step (ii). For example, BenAri 's formal implementation of structures associated with state – e.g. annotations like labeling or tableaus/structure, marking, rank, predicates choice -- for each possible transitions in the product process UxB (*Semantics for $\mathcal{UB}$: infinite path* - pg. 210) wherein elements of two spaces are intersected as set forth above. As for cross-product or intersection, intersection is an operation constructing a finite state abstraction machine

representing the common transition sequences of two such finite state machines, and is defined as the usual cross-product construction of states and transitions. BenAri's propositional calculus teaches using structures (structure H, Tableau - pg. 214; Ranking - pg. 217; Marking pg. 218) to support proof for completeness (or correctness) based on some propositional theorem or axiom. In Automata theory, evaluating the annotations of two states being combined by conjunction or association to this cross-product of the two states will determine a truth value associated with the state corresponding to said annotations, conditions or abstract meta-structure, in terms that a final state is evaluated as true when on the state reachable by the corresponding transition based on the associated annotations implicated in a given cross-product. Based on proposition and proving of correctness, certain branch paths or transitions are resolved in terms of transitions proven valid as opposed to having a virtually infinite number of non-deterministic transitions from the onset. As in BenAri (see § 3-6, pg. 211-224), given the large complexity in a possibly infinite amount of transitions, BenAri teaches propositional calculus for formalizing a set of conditions or propositions by way of formal abstracted expressions, and via proof by induction, validating whether certain conditions hold true, thereby validating correctness of a set of requirements regarding transitions. From a very large time complexity, proof by induction based on some model and very selective propositions to validate, BenAri has been able to narrow down a program branching time complexity to a simpler and verifiable model. As well established in the theoretical study of computer program for analyzing time complexity, many endeavors (as in BenAri) to reducing this complexity was well known, hence the concept of reducing is therefore **integral to** the result provided from the above proof of correctness using automata theory, when one contemplates the huge amount of unproven path or transitions of a program regarding all the

non-deterministic combinations for branching (see BenAri: see *branching time approach, correct termination for ... one possible computation for every input ... study the property one possible computation ... all possible choices* - middle pg. 208; *Semantics for $\mathcal{UB}$: infinite path* - pg. 210).

(E) One of ordinary skill in the art would construe the $\mathcal{UB}$ model (see BenAri: Introduction, pg 208) by which BenAri's calculus or formal propositions are formally implemented to be a algorithm using variety of predicates or choices leading to reducing a large complexity; which can be analogized to the cross-product of a program input and a branching property expressed as a automaton as this is laid out in recited steps (i) (ii) and (iii) from above. In other words, how to perceive reduction by BenAri's approach can be construed as follows: from the original large number of possible transitions/states, some proof has been fulfilled to render non-deterministic transitions to become deterministically valid, thereby reducing the amount of non-deterministic state transition from the outset. That is, BenAri's formalizing and computing of branching transitions as from above is deemed fulfilling the claim scenario taken as steps (ii) and (iii). BenAri's describing of a product (*$\mathcal{UB}$: unified model for a branching time* – pg. 208, bottom) by which all possible combinations of a program are being explored does fulfill the cross-product for a branching time property B, i.e. applying intersection of an input program with a automaton space expressing a branching time property reads on cross-product as recited in (i). From all possible combinations from BenAri's cross-product, the reduced number of states is deemed the end-result of BenAri's calculus approach for establishing correctness in regard to specific transition (§ 3-5, pg. 211-220) of the program based on the original amount of all possible combinations state/transactions using some theorems/axioms and completeness proof whereby

validated transitions or behaviors are correct as opposed to a much larger amount of non-deterministic combinations as set forth above. When the claim does not teach **how** the reducing is done, BenAri's way of determining which transitions can be proved correct is deemed sufficient to entails a representation of the branching property expressed as a cross-product in which number of state/transitions are less voluminous as from the onset, as set forth in sections C and D. Appellant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

(F) Appellant has submitted BenAri does not disclose or suggest *preserving* a branch time property (Appeal Brief pg. 6, lines 24-26). Any feature provided in the preamble of the claim (i.e. *A method for reducing ... that preserves at least one branching time property*) will bear its weight only to the extent at which such feature has been elaborated in the body of the claim, or particularly specified in the claim as a whole. The claimed steps recited as in (i) (ii) (iii) does not make a remote linkage to how a 'branch time property f' is preserved. Absent any details in the claim as far as how exactly preserving is done, the 'preserves ... one branching time property' has been treated in terms of the steps actions recited as (i) (ii) and (iii) and in that light, addressed as set forth in the rejection. As set forth in the above sections B to E, the recited steps (i) to (iii) have been deemed matched by BenAri. For one of ordinary skill in the art with knowledge of *formal methods* in the art of solving or find the best solution to the time complexity of a program by way of verifying of conditions as from the teachings by BenAri, property preserving is done via using mathematical abstractions, whereby given a program and a property to be verified (branching time property), finding a simpler abstract program such that satisfaction on the

abstract program implies satisfaction on the initial program. BenAri applies different mathematical constructs and equations to set up properties in terms of formalized propositions with conditions to be verified by proof, whereby, given a initial property, when some abstract and simplified mathematical representation of the initial concrete program is proved to be inductively correct, said property thus abstracted from the initial program is preserved as per its correctness. The above argument amounts to allegation with insufficient grounds as to show how the teachings by BenAri do not meet the language of the claim.

(G) Appellant has submitted that BenAri fails to teach or suggest 'forming a product of the program M and a said branching time property' (Appeal Brief pg 7, top line). BenAri teaches a product $\mathcal{K}\mathcal{B}$ which is a product of a unified system (automaton space for a program) with a branching time property of such program, abstracted as a model to represent *all possible computations of a program* (bottom, pg. 208), using formal methods or propositional calculus whereby all properties (universal or non-deterministic – see top pg 209) of a program are formalized via symbols, structures annotations, formulas, theorems needed (§ 2 pg. 210-212) to effectuate the inductive proof in regard to correctness of conditions being proposed. As proffered in the SUMMARY OF CLAIMED SUBJECT MATTER (Appeal Brief pg. 2), the 'forming a product M' and 'branching time property f expressed as an automaton' is disclosed in the Disclosure at page 6, lines 18-22. This pointed to portion of the Disclosure mentions about alternating transition system (ATS) where soundness is shown by constructing a valid concrete proof of correctness given feasibility of the abstract ATS, e.g. if M = (S, I, R, L) and A = (Q ... F) such that M x A be the product of this ATS which is shown in step 210 of Figure 2; which, at a closer look, is not clearly establishing where branching time property f is intertwined to the

automaton A as claimed. In view of the insufficient teaching as to how a branching time f is expressed as automaton A, this disclosed section can be tentatively interpreted as following: a program is expressed in formalized expressions for some properties, and some property space/transitions is expressed as A, such to formalize a mathematical representation of M intersected with A, a cross-product is exhibited as M x A. However, what is important here is that BenAri also teaches intersection of a program M in view of branching time expressed each as U and B in a form of product (*$\mathcal{U}\mathcal{B}$: unified model for a branching time* – pg. 208, bottom). Based on what appears to be Appellant' vague mapping between the above section of the Disclosure and the claimed 'forming a product' M x A, and well-known concepts as set forth in section D, it is deemed that interpretation of this 'forming a product' limitation has been fulfilled in BenAri forming of the product UxB (*unified model for a branching time*). Appellant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

(H) Appellant has submitted that BenAri fails to teach or suggest 'abstract domain containing a set of abstract values to generalize possible states of said program' (Appeal Brief, pg. 7 lines 3-5). A domain containing a set of abstract values to generalize possible states of a program is the very calculus by BenAri using formal methods and formalizing of concrete program by way of cross-product as set forth in sections B-C. BenAri does teach formal methods to abstract behavior of branching time (see Introduction, pg. 208) in regard to a input program using a unified model (see pg. 208, bottom) for which set of values implemented via calculus form of expressions (see § 2-5, pg. 211-220; Fig. 1-2) to formulate propositions and conditions which

are set for the ensuing proof of soundness/correctness thereof -- proof in that satisfaction of the abstracted representation of the branch property is fulfilled - thereby safeguard the validity of a branching property expressed as a possible transaction from a state. The claim language teaches a set of abstract values to generalize possible states and this is deemed disclosed in BenAri (see § 2-5, pg. 211-220; Fig. 1-2) whereby state and transitions are abstracted via formulation of symbols, expressions, operators and theorem, axiom/lemma constructs. The lack of further details in the claim language cannot preclude the teachings by BenAri from reading into the claim as it is interpreted.

(I) Appellant has submitted that BenAri fails to teach or suggest 'computing an abstract program with reduced number of states' and 'an altered version of said branch time property f' (Appeal Brief, pg. 7, lines 6-13). The approach by BenAri is to formalize certain calculus propositions based on a selected model and implement induction proof for verifying of some selective propositions by formulating abstract constructs, thereby putting into consideration the branching time of the input program in terms of structures, axiom, expressions, conditions, tuples and set of conditions based on which BenAri provides proof of correctness so as to justify whether a proposed or to-be-verified property holds true; all of which has been explained in sections D-G from above. The reduction of an otherwise large number of states is deemed integral to the proposition calculus by BenAri, because only those selected propositions when established as sound or correct are valid, leaving out those which are not proven correct; the concept of *reducing* can also be referred back to sections D and E from above. The claim language does not establish clear teachings about what exact mechanism 'reduced states' is being derived from; nor does it clearly provide details about how a version of branch time is introduced

in order for another version thereof is altered via explicit action step. The preserving of a property has been interpreted as BenAri's being able to establish proof (of soundness) that a transaction from a given state including property for that state is correct by induction proof, because otherwise all proposed settings or logical predicates (e.g. axioms, theorems, lemma) or meta-annotation (e.g. Structure H, Tableaux, marking – see pg. 214-216) regarding a branching time property expressed as a transaction would be discarded if invalid. The broad language of the claim coupled with the lack of action steps interrelating abstract domain, set of abstract values as in (ii), and product M by A as in (i), on the one hand, and *abstract program* (as in iii) with reduced number of states and altered version of f, on the other hand cannot preclude the approach using propositional calculus of BenAri from meeting the claim as it is interpreted, and understood according to level of one of ordinary skill in the art pertinent to Automata Theory, as set forth in sections B through F. Appellant has remarked that the Examiner has not shown how formulas and axioms read on (*) *elimination of paths*, and that Examiner has made no attempt to indicate how (**) *branching time property is altered to form the altered version of said property*; that (***) BenAri does not address reducing a program 'while preserving' branching time properties (Appeal Remarks, pg. 7, lines 10-17). The claim as recited does not provide sufficient details as to enforce any such alleged requirement as raised in (*) and (**). Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference. As for (***), first, the 'reducing while preserving' is not remotely commensurate with the claim language. Second, BenAri' s calculus is to abstract the concrete space of the target program in function of its branching time

property in terms of a judicious choice of formal model called an unified model ($\mathcal{U}\mathcal{B}$: *unified model for a branching time* – pg. 208, bottom), wherein branching time property for each transition is expressed as abstracted calculus (or formal methods) logical axiom, expressions with data structures to set up conditions for transitions given a initial state. The proof by induction based on predicates or axiom, theorem or to-be-verified conditions will validate whether those conditions hold true, hence preserving the pertinent branch property to that effect; all of which has been set forth in section F above.

The arguments are not persuasive because it is believed that the Office Action has established a clear prima facie case of rejection; therefore, the claims 1-7, 12-20, 25-26 will stand as set forth in the Office Action.

----

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Tuan A Vu/

Primary Examiner, Art Unit 2193

Conferees:

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

/Eddie C Lee/

Supervisory Patent Examiner, TC 2100